Journal of Computer and Knowledge Engineering, Vol. 7, No. 1, 2024. (1-8)

Ferdowsi University of Mashhad

Journal of Computer and Knowledge Engineering

https://cke.um.ac.ir

Information and Communication Technology Association of Iran

ENIXMA: ENsemble of EXplainable Methods for Detecting Network Attacks*

Research Article

Seyed Mojtaba Abtahi[1], Hossein Rahmani[2], Milad Allahgholi[3], Sajjad Alizadeh[4]

DOI: 10.22067/cke.2024.82986.1084

Abstract: The Internet has become an integral societal component, with its accessibility being imperative. However, malicious actors strive to disrupt internet services and exploit service providers. Countering such challenges necessitates robust methods for identifying network attacks. Yet, prevailing approaches often grapple with compromised precision and limited interpretability. In this paper, we introduce a pioneering solution named ENIXMA, which harnesses a fusion of machine learning classifiers to enhance attack identification. We validate ENIXMA using the CICDDoS2019 dataset. Our approach achieves a remarkable 90% increase in attack detection precision on the balanced CICDDoS2019 dataset, signifying a substantial advancement compared to antecedent methodologies that registered a mere 3% precision gain. We employ diverse preprocessing and normalization techniques, including z-score, to refine the data. To surmount interpretability challenges, ENIXMA employs SHAP, LIME, and decision tree methods to pinpoint pivotal features in attack detection. Additionally, we scrutinize pivotal scenarios within the decision tree. Notably, ENIXMA not only attains elevated precision and interpretability but also showcases expedited performance in contrast to prior techniques.

Keywords: Network anomaly detection, Machine learning, Intrusion detection system, Ensemble learning, Interpretability.

1. Introduction

Presently, the Internet constitutes a principal component of society. Given the pervasive nature of the Internet, its accessibility is viewed as indispensable. Conversely, attackers strive to disable Internet services and exploit Internet service companies [7]. One of the most common attacks these companies face is DDoS attacks, which disrupt their service provision. Service disruptions and outages can inflict substantial damages on a company, to the extent that a 24-hour service outage in a major e-commerce company can result in millions of dollars in losses [1].

Network traffic can be divided into two categories: normal traffic and DDoS attack traffic. Based on traffic characteristics, it can be determined when an attack is being perpetrated on the victim network [6]. DDoS attacks are typically volume-based, and flow-based methods are suitable for detecting these types of attacks. Flows are defined as a collection of IP packets that pass through a specific point in the network within a certain time interval.

In such a way that packets associated with a particular flow exhibit shared characteristics. There are three stages to attack detection: in the first stage, flow exporters receive and aggregate raw packets. In the subsequent stage, flow collectors store and preprocess flow data. Finally, in the last stage, parsing and analysis programs, such as Intrusion Detection Systems (IDS), retrieve and analyze flow data [2,3].

Numerous algorithms such as KNN, SVM, and RF are utilized in Intrusion Detection Systems. These algorithms make decisions based on the features they obtain from the input data [4,5]. Given the high importance of interpretability in network-related data for better attack detection, there has been a fundamental need for interpretability in machine learning issues related to attack detection [8].

In the second section, we will review the work done in the field of anomaly detection in network data using different approaches and review the work done in the field of interpretability and important algorithms for interpretability. In the third section, we will elaborate on the different parts of the proposed method and the outputs of interpretability algorithms. In the final section, we will assess the results of the proposed method and discuss the reasons for its superiority compared to previous works [31].

* Manuscript received: 2023 June 17, Revised, 2023 August 18, Accepted, 2024 May 1.

[1] M.Sc. School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.

[2] Corresponding author. Assistant Professor, School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran. **Email:** h_rahmani@iust.ac.ir

3 Ph. D. Student School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.

4 MSc. School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.

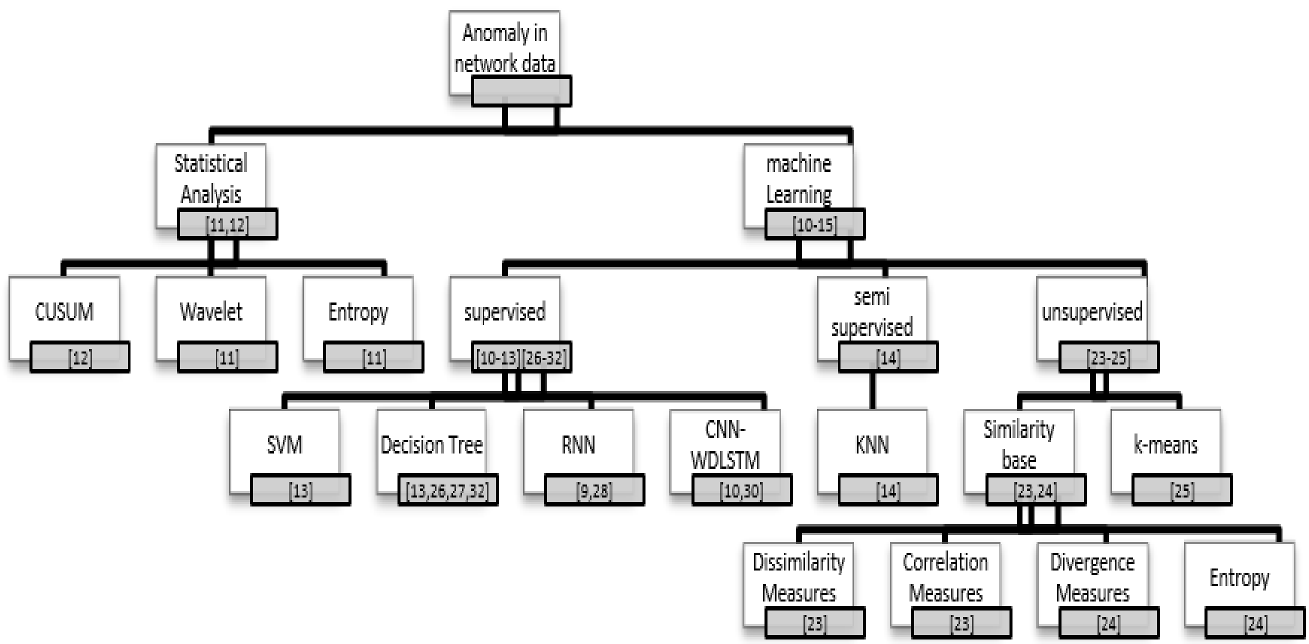


Figure 1. A General Overview Of The Categorization Of Tasks Performed In The Field Of Anomaly Detection In Network Data

2. Previous Studies

In this section, we will first delve into some of the work done in the field of anomaly detection in network data using machine learning algorithms and similarity metrics. In the second part, we will review studies related to interpretability.

2.1. Anomaly Detection in Network Data

Research carried out in this field can be examined from two aspects: 1- Statistical analysis, 2- Machine learning. A general schematic of the metrics and methods mentioned in this study can be seen in Figure (1).

Statistical Analysis. In this section, we delve into the various solutions for detecting anomalies in the network. Anomaly detection methods can be divided into two categories: statistical analysis and machine learning. Statistical analysis-based methods require relatively less computational power. Although this approach offers a rapid and acceptable detection rate, its main issue is the false-positive rate.

Girma and colleagues [11] propose a combined statistical model that can significantly reduce these attacks and can be a better alternative solution for current detection problems. This combined scheme is based on entropy matrices and covariance. The advantages of this method include high accuracy and independence from any assumptions in network packets. One of the drawbacks of this method is the decrease in focus of this scheme during the aggregation of calculations.

Rudolf and colleagues [12] use a statistical analysis method that operates on multi-layered network traffic. Two methods are used in this paper. Both methods use a threshold value to detect anomalies in the network. In the first method (1), Ssprt[t] is the test variable. P0(Npk[t]) and P1(Npk[t]) are the probability density functions before and after the attack. If Ssprt[t] is greater than the threshold value of tr, there is a probability of an attack.

$$S_{sprt}[t] = max\left\{0, [S_{sprt}[t-1] + \log(\frac{P_1(N_{pk}[t])}{P_0(N_{pk}[t])})]\right\}; \quad S[0] = 0 \tag{1}$$

In the second method (2), the difference between the current average and the long-term average is calculated. (Npk[t] is the average traffic at time t, and m[t] is the long-term average up to time t). When the cusum coefficient exceeds the threshold, there is a probability of a DDoS attack.

$$S[t] = max\{0, (S[t-1] + N_{pk}[t] - m[t])\}; S(0) = 0 \tag{2}$$

There are three appealing features in these approaches. First, both methods are self-learning, which allows them to adapt to the network and its patterns. Second, these methods reduce the average detection time of attacks. Third, they are computationally simple and can therefore be implemented linearly. Fourth, the rate of false positives or false detection is low [13].

Machine Learning. The next category includes machine learning-based systems that employ data mining techniques to discover unknown algorithms in large volumes of data. Machine learning systems use different approaches in their systems, including supervised, semi-supervised, and unsupervised approaches.

In the supervised approach, the training data must be

labeled as attack and non-attack cases, which is very time-consuming and may encounter unintended errors.

In the semi-supervised approach, the training dataset doesn't need to be fully labeled. Although it reduces the complexity of labeling, it increases the ambiguity of the model providing network or system traffic.

Unsupervised approaches do not require labels. These systems cluster similar patterns and behaviors.

Chuanlang and colleagues [9] use recurrent neural networks to detect anomalies. They use both forward and backward propagation methods in their methodology. Their experiments are performed on the KDD-NSL dataset [30]. The classification is based on whether the attacks are normal or not. In their experiments, they increased the features from 41 features to 122 features, so the RNN-IDS model has 122 input nodes and 2 output nodes in binary classification experiments. The number of epochs is also 100. They performed the experiments with the number of hidden nodes, 20, 60, 80, 120, 240, and the learning rate, 0.01, 0.1, 0.5. The highest accuracy is for the number of hidden nodes 80 and the learning rate of 0.1.

Abhijit Das and colleagues [10] used a combined approach based on three models: Balanced Bagging, XGBoost, and RF-HDDT. The parameters of Balanced Bagging and XGBoost are tuned for imbalanced data, and the Hellinger criterion complements the Random Forest to overcome the limitations of the default distance criterion. They propose two new algorithms to address the issue of class overlap in the dataset and apply them during training. These two algorithms are used to help improve the performance of the test dataset by influencing the final classifier decision made by the three basic classifiers as part of the ensemble classification, which uses a majority vote combiner.

Their proposed scheme performs noticeably better than reported schemes for binary and multi-class classification cases. This implies that their combined approach can effectively handle both binary and multi-class anomaly detection problems, offering an advantage over traditional methods.

Hua-Wu and colleagues [14] initially examine the architecture of DDoS and ascertain the details of its stages. They then study the procedures of DDoS attacks and select variables based on these characteristics. Ultimately, they use the K-nearest neighbor method to classify the network status at each stage of a DDoS attack. As you can see in Figure 2, the process works such that after training the K-nearest neighbor algorithm based on 9 selected features, data is collected online, then preprocessed, and in the final stage, it is classified into three classes: normal traffic, attack traffic, and pre-attack traffic. After conducting an experiment on this algorithm, the accuracy of this algorithm on the 2000 DARPA dataset is 91%.

This result suggests that their approach, using the K-nearest neighbors algorithm with selected features that are representative of DDoS attacks, is effective in identifying such attacks. It is noteworthy that the accuracy of this method is quite high, indicating that it could be a robust method for real-world network security applications.

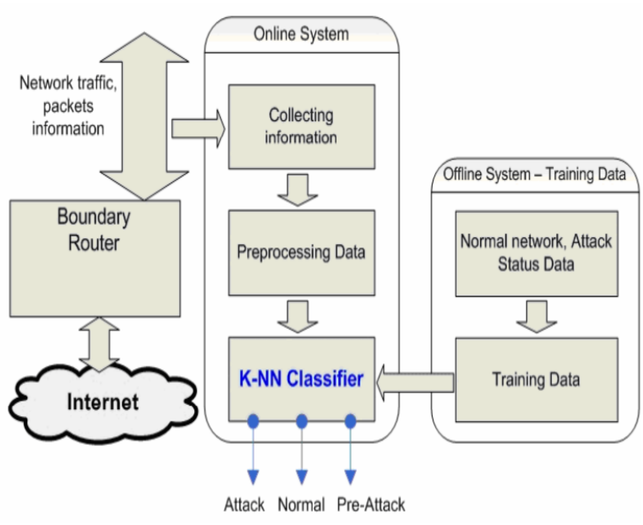


Figure 2. General overview of the classification process using the k-nearest neighbors algorithm [14]

Lazarovich and colleagues [15] focused on comparing anomaly detection techniques in unsupervised algorithms. They pursued various schemes for detecting outlier data for self-anomaly detection in their work. Most anomaly detection algorithms need a completely normal set of data to train the model and implicitly assume that anomalies can be identified as patterns that have not been previously observed. As an outlier might impact the measurements and modeling, we need to consider different plans for extracting these data to understand which one works effectively [16].

Their research emphasizes the importance of proper data handling, especially outliers, in the training process of anomaly detection models. By using various schemes for handling outlier data, they sought to determine the most effective methods for improving the model's ability to accurately identify anomalies. The results of such studies can significantly contribute to the enhancement of unsupervised learning algorithms used for anomaly detection, particularly in the context of network security and DDoS attack identification.

Bolodurina and colleagues [32] investigate the issue of improving the accuracy of classification of network attacks on unbalanced CICDDoS2019 data using class sampling algorithms such as ROS, SMOTE, and ADASYN. The results of computational experiments show the effectiveness of data balancing algorithms in identifying network attacks. Additionally, the ADASYN adaptive synthetic sampling method improves the accuracy of type of attack classification by up to 84% compared to other algorithms.

2.2. Interpretability

Among the significant methods in interpretability are LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) which have been extensively used in various works [17]–[20]. These

methods provide non-intuitive, local, and model-independent interpretability.

In a study, Rizi and colleagues [18] used the LIME method to examine the prediction performance of an LSTM (Long Short-Term Memory) model. They focused on extracting significant features in samples with incorrect predictions and, ultimately, by altering the effect of negative parameters, they improved the accuracy of their model.

In a similar study, Singgata and colleagues [19] investigated the important features in the output of the XGBoost model. They implemented their findings on a dataset of user logs.

They implemented their findings on a user log dataset. However, a fundamental point that exists in all similar articles is that there is no evaluation for interpretability. Marcelo and colleagues [22] have used interpretability for feature extraction. They used the SHAP method and, based on the score this method considers for each feature, they proceeded to select important features. Then, after feature selection, they carried out the classification task. In this study, the results of SHAP are compared with other existing feature selection methods such as ANOVA (Analysis of Variance). The results indicate that SHAP performs better in feature selection compared to other methods.

3. proposed method

In this section, we examine our proposed method in this paper. As you can see in Figure 3, we have used a combined method to detect the type of attacks, and in addition, we use interpretability algorithms to identify important features, distinguishing characteristics for identifying the type of attack, and extracting important rules for detecting the type of attack.

3.1. Dataset

In this paper, we use the CICDDoS2019 dataset. This dataset includes 7 types of DDoS attacks, namely LDAP, MSSQL, NetBIOS, Portmap, Syn, UDPLag, UDP, in CSV format. Additionally, this dataset includes 88 features. The dataset comprises 20,364,532 data records. This is a combined dataset of volumetric features in network packets, header features of packets, and statistical features on volumetric features [21].

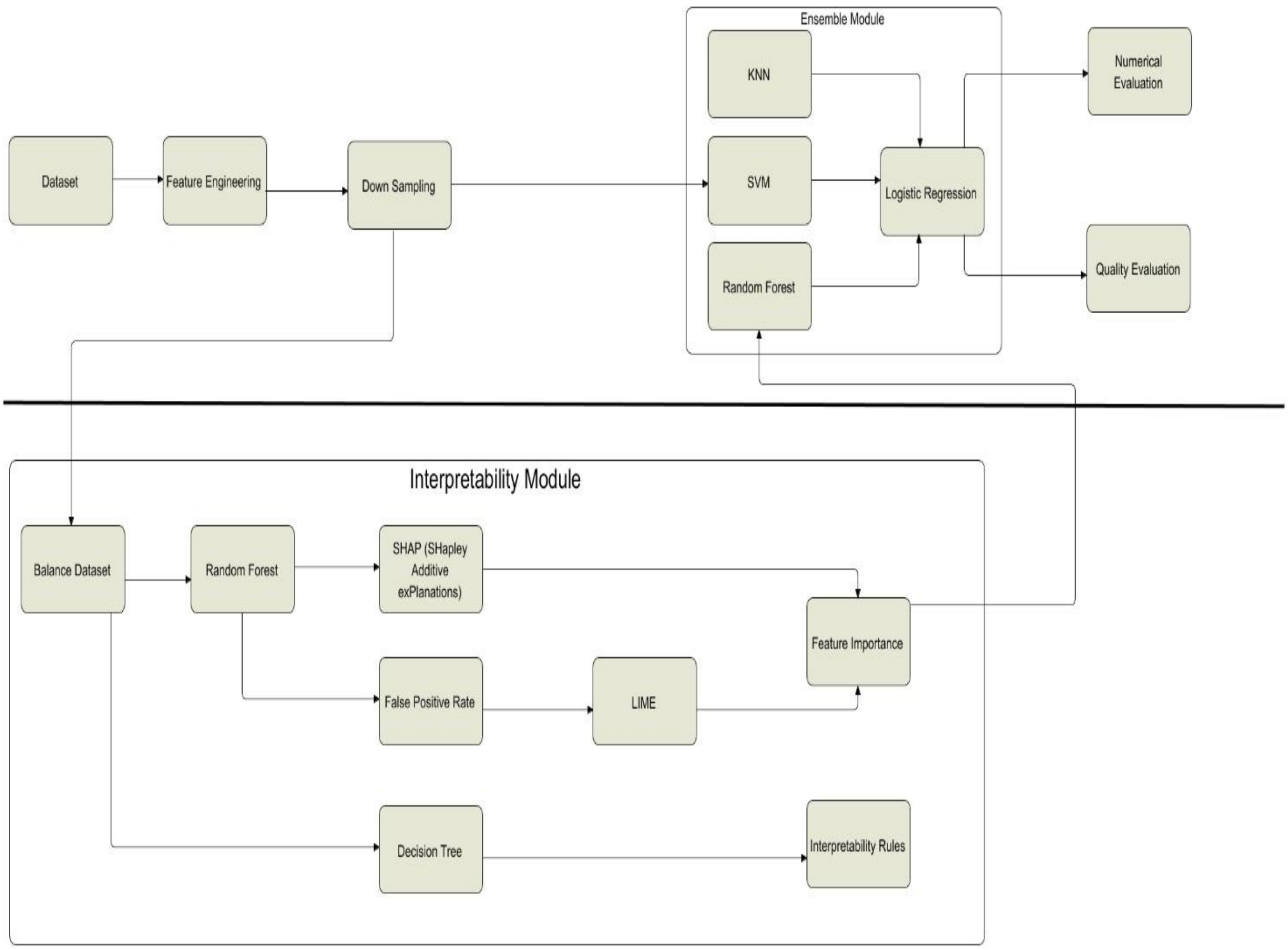


Figure 3. The different stages of the interpretive process in the proposed method

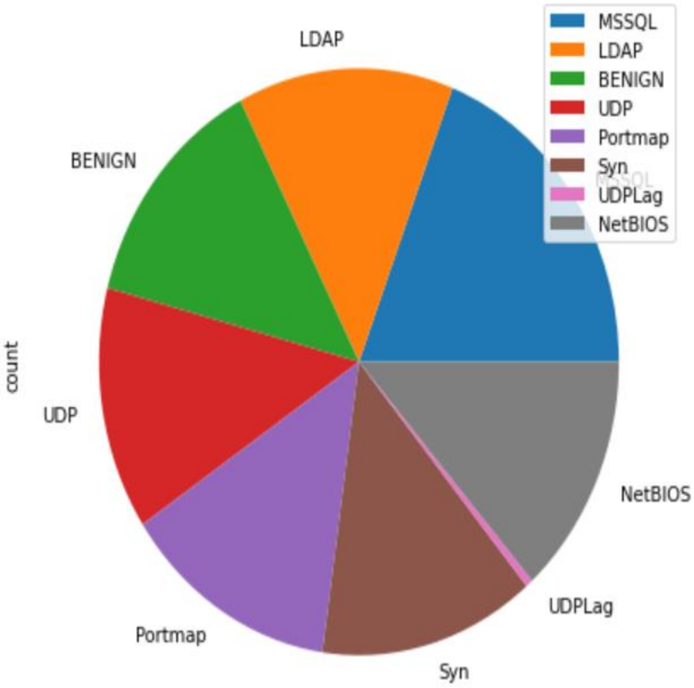


Figure 4. The distribution of labels after balancing the dataset.

3.2. Preprocessing

Initially, we remove missing values from the dataset, then we remove missing and single value data in all data. After that, we replace 'inf' values due to the fact that they impede the correct training of the model, with the maximum value of that feature. In the next step, using the z score normalization, we remove outlier data from the dataset. Finally, to better train the model, we normalize the data.

3.3. Balancing the Dataset

Due to the imbalance of the data and the high ratio of datawith the attack label to data with the non-attack label, sampling was performed on the existing data, and the distribution of labels became as shown in Figure 4. Furthermore, we employed the undersampling technique to balance the dataset, using the clustering centroids method as our approach for selecting samples to be removed [26]. Due to the low number of data with the UDPLag label, even after balancing the dataset, the number of these data is still lower compared to the rest of the labels.

3.4. Interpretability Module

In this section, we used three algorithms, SHAP, LIME, and decision tree, for better interpretability of attack detection and identification of important features for attack detection.

SHAP Algorithm. The SHAP algorithm is a interpretation method used to analyze the influence of features on the predictions of a model. This algorithm is built upon cooperative game theory and Shapley value [17]. In this algorithm, for each instance of the data, the impact of each feature on the model's output prediction is calculated. To compute this influence, first, the Shapley value is calculated for each combination of features. Then, using these values, the impact of each feature on the model's prediction is calculated [22]. In our method, we first feed the balanced dataset into the random forest algorithm, and then we extract the important features using the SHAP algorithm.

LIME Algorithm. The LIME algorithm is a model interpretability algorithm that is used to understand the behavior of machine learning models. This algorithm is particularly intended for models that operate in a non-linear manner. LIME is an acronym for Local Interpretable Model-agnostic Explanations and it essentially provides explanations for each input data by interpreting the model's decisions using a local linear function, stating which features are important for decision-making [20]. In our method, we first feed the balanced dataset into the random forest algorithm, then we separate the false positive data and give it to the LIME algorithm, and then we isolate the important and influential features.

Decision Tree. Extracting rules from a decision tree can be very beneficial and vital in many machine learning models. Here are some explanations about the importance of extracting rules from a decision tree:

1. Transparency and interpretability: A decision tree, as a machine learning model, can be very complex and structural. Extracting rules from a decision tree can help users fully understand the structure and operation of the tree and make better decisions.
2. Reducing costs and time: Extracting rules from a decision tree can help users who spend more time and costs on training machine learning models in larger and more complex programs. With the extracted rules,

more data can be processed easily and more trust can be placed in the trained model.

3. Increasing accuracy and improving performance: Rules extracted from a decision tree can help users improve the performance of the machine learning model. Using these rules, better predictions and decisions can be made directly.
4. Helping network admins: The extracted rules can help network admins to identify attacks easier using patterns and rules. In our method, we first implement the decision tree on our data, then we extract repeating patterns from our tree.

4. Results

In this paper, we proposed a combined method of interpretable models based on SHAP and LIME algorithms, and we selected the best features, which are a combination of packet volume features and flag count features. In this paper, we performed two types of evaluations: quantitative evaluation and qualitative evaluation.

4.1. Quantitative Evaluation

In this segment, a quantitative evaluation of the proposed methodology is carried out. Initially, the critical features incorporated within the algorithms utilized in the proposed approach are thoroughly investigated. The following represent the 15 most significant features as per the SHAP and LIME algorithms:

Table 1. Important features of the dataset based on LIME and SHAP algorithms

SHAP Feature Selection	LIME Feature Selection
Fwd Act Data Pkts	Packet Length Std
Total Fwd Packet	Init Win bytes forward
Flow Bytes/s	Packet Length Variance
Fwd Packet Length Max	Min Packet Length
Fwd Packet Length Std	ACK Flag Count
Packet Length Mean	URG Flag Count
Total Bwd packets	Fwd Packet Length Min
Flow duration	Total Backward Packets
Bwd IAT Std	Subflow Bwd Packets
Bwd Packet Length Max	Bwd Packets/s
Bwd Packet Length Mean	SYN Flag Count
Total Length of Bwd Packet	Fwd Packet Length Mean
Total Length of Fwd Packet	Fwd PSH Flags
Protocol	RST Flag Count
Fwd Packet Length Min	Average Packet Size

Most of the features selected based on the SHAP algorithm are volumetric features, and most of the features selected based on the LIME algorithm are features based on the number of flags.

Quantitatively, as you can see in Table 3, our method increases the accuracy and also due to the reduction in the number of features, it decreases the execution time.

Table 2. Quantitative evaluation of implemented algorithms in the article

Algorithm Name	Accuracy	Precision	Recall	F1-score
Random Forest [32]	84%	83%	83%	82%
Ensemble Method (RF+SVM+KNN)	87%	88%	83%	86%
Ensemble Method + SHAP	87%	87%	88%	86%
Ensemble Method + ENIXMA	89%	90%	89%	88%

4.2. Qualitative Evaluation

In this section, we address recurring scenarios in the decision tree.

Scenario 1:

```
|   |   |   |---  act_data_pkt_fwd <= 0.50
|   |   |   |   |---  Total Backward Packets <= 33.50
|   |   |   |   |   |--- class: BENIGN
|   |   |   |   |---  Total Backward Packets >  33.50
|   |   |   |   |   |--- class: Syn
|   |   |   |---  act_data_pkt_fwd >  0.50
|   |   |   |   |---  Fwd Packet Length Std <= 0.28
|   |   |   |   |   |--- class: Syn
|   |   |   |   |---  Fwd Packet Length Std >  0.28
|   |   |   |   |   |--- class: BENIGN
```

In scenario 1, we observe the impact of three features, act_data_pkt_fwd, Total Backward Packets, and Fwd Packet Length Std, in detecting a SYN attack from non-attack data.

Scenario 2:

```
|   |---  Fwd Packet Length Min >  118.50
|   |   |---  Fwd Packet Length Max <= 319.50
|   |   |   |---  Flow Duration <= 44.50
|   |   |   |   |--- Flow Bytes/s <= 276000000.00
|   |   |   |   |   |--- class: Portmap
|   |   |   |   |--- Flow Bytes/s >  276000000.00
|   |   |   |   |   |--- class: Portmap
|   |   |   |---  Flow Duration >  44.50
|   |   |   |   |--- Flow Bytes/s <= 2083567.06
|   |   |   |   |   |--- class: BENIGN
|   |   |   |   |--- Flow Bytes/s >  2083567.06
|   |   |   |   |   |--- class: NetBIOS
```

In scenario 2, two flow features, Flow Duration and Flow Bytes/s, play a significant role in detecting Protmap and NetBIOS attacks.

Scenario 3:

```
|   |---  Fwd Packet Length Max >  319.50
|   |   |   |--- Total Length of Fwd Packets <= 737.00
|   |   |   |   |---  Fwd Packet Length Max <= 364.50
|   |   |   |   |   |--- class: UDP
|   |   |   |   |---  Fwd Packet Length Max >  364.50
|   |   |   |   |   |--- class: UDPLag
|   |   |   |--- Total Length of Fwd Packets >  737.00
|   |   |   |   |---  Fwd Packet Length Mean <= 399.50
|   |   |   |   |   |--- class: UDP
|   |   |   |   |---  Fwd Packet Length Mean >  399.50
|   |   |   |   |   |--- class: UDP
```

In scenario 3, we observe that the high volume features are the reason for detecting UDP attacks.

From a qualitative perspective, we were able to identify scenarios and important features for detecting types of attacks. This helps network experts to better identify and detect various attacks. Furthermore, in accordance with the article by Wei Gao et al. [33], the packet volume magnitude in detecting UDP attacks and the length of transmitted packets are crucial features in detecting Syn attacks.

5. Conclusion

With the increasing use of the Internet, identifying and then hardening the network against DDoS attacks is one of the main goals of network administrators. In this paper, while reviewing previous work on attack detection, we introduced a new method called ENIXMA for identifying and interpreting attacks carried out on the network. Quantitatively, ENIXMA has led to a 3% improvement in the accuracy of previous methods. Moreover, by taking advantage of interpretability methods in machine learning, ENIXMA not only identifies attacks but also describes the reasons for the occurrence of the attack and the features that have an impact on that type of attack. Interpreting the attack provides the network administrator with the opportunity to take appropriate action to harden the network. For future work, it is suggested to identify other network attacks such as port scanning and to take advantage of more advanced interpretability methods.

6. References

[1] M. Aamir and S. M. Ali Zaidi, "Clustering based semi-supervised machine learning for DDoS attack classification," Journal of King Saud University - Computer and Information Sciences, vol. 33, no. 4, May 2021, doi: 10.1016/j.jksuci.2019.02.003.

[2] S. Zavrak and M. Iskefiyeli, "Anomaly-Based Intrusion Detection From Network Flow Features Using Variational Autoencoder," IEEE Access, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3001350.

[3] R. Bhatia, R. Sharma, and A. Guleria, "Anomaly Detection Systems Using IP Flows: A Review," 2021. doi: 10.1007/978-981-16-0235-1_80.

[4] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," Information Sciences, vol. 513, Mar. 2020, doi: 10.1016/j.ins.2019.10.069.

[5] S.-T. Chiu and F.-Y. Leu, "Detecting DoS and DDoS Attacks by Using CuSum Algorithm in 5G Networks," 2021. doi: 10.1007/978-3-030-57811-4_1.

[6] M. Nooribakhsh and M. Mollamotalebi, "A review on statistical approaches for anomaly detection in DDoS attacks," Information Security Journal: A Global Perspective, vol. 29, no. 3, May 2020, doi: 10.1080/19393555.2020.1717019.

[7] S. Hosseini and M. Azizi, "The hybrid technique for DDoS detection with supervised learning algorithms," Computer Networks, vol. 158, Jul. 2019, doi: 10.1016/j.comnet.2019.04.027.

[8] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," Commun. ACM, vol. 63, no. 1, pp. 68–77, 2020, doi: 10.1145/3359786.

[9] C. Yin, Y. Zhu, J. Fei, and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," IEEE Access, vol. 5, pp. 21954–21961, Oct. 2017, doi: 10.1109/ACCESS.2017.2762418.

[10] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," Information Sciences, vol. 513, pp. 386–396, Mar. 2020, doi: 10.1016/j.ins.2019.10.069.

[11] A. Girma, M. Garuba, Jiang Li, and Chunmei Liu, "Analysis of DDoS Attacks and an Introduction of a Hybrid Statistical Model to Detect DDoS Attacks on Cloud Computing Environment," Apr. 2015. doi: 10.1109/ITNG.2015.40.

[12] R. B. Blažek, H. Kim, B. Rozovskii, and A. Tartakovsky, "A novel approach to detection of 'denial-of-service' attacks via adaptive sequential and batch-sequential change-point detection methods," 2001.

[13] S. R. Gaddam, V. v Phoha, and K. S. Balagani, "K-Means+ID3: A Novel Method for Supervised Anomaly Detection by Cascading K-Means Clustering and ID3 Decision Tree Learning Methods."

[14] Hoai-Vu Nguyen and Yongsun Choi, "Proactive Detection of DDoS Attacks Utilizing k-NN Classifier in an Anti-DDos Framework," World Academy of Science, Engineering and Technology , 2010.

[15] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection," May 2003. doi: 10.1137/1.9781611972733.3.

[16] C.-K. Han and H.-K. Choi, "Effective discovery of attacks using entropy of packet dynamics," IEEE Network, vol. 23, no. 5, Sep. 2009, doi: 10.1109/MNET.2009.5274916.

[17] C. Di Francescomarino and C. Ghidini, "Predictive Process Monitoring," in Lecture Notes in Business Information Processing, 2022, vol. 448, pp. 320–346. doi: 10.1007/978-3-031-08848-3_10.

[18] W. Rizzi, C. Di Francescomarino, and F. M. Maggi, "Explainability in predictive process monitoring: When understanding helps improving," in Lecture Notes in Business Information Processing, 2020, vol. 392 LNBIP, pp. 141–158. doi: 10.1007/978-3-030-58638-6_9.

[19] R. Sindhgatta, C. Ouyang, and C. Moreira, "Exploring interpretability for predictive process analytics," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2020, vol. 12571 LNCS, pp. 439–447. doi: 10.1007/978-3-030-65310-1_31.

[20] D. Adi and N. Nurdin, "Explainable Artificial Intelligence (XAI) towards Model Personality in NLP task," IPTEK J. Eng., vol. 7, no. 1, p. 11, 2021, doi:

10.12962/j23378557.v7i1.a8989.

[21] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, “Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy,” in Proc. 53rd International Carnahan Conference on Security Technology, Chennai, India, 2019.

[22] W. E. Marcilio and D. M. Eler, “From explanations to feature selection: Assessing SHAP values as feature selection mechanism,” in Proceedings - 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2020, 2020, pp. 340–347. doi: 10.1109/SIBGRAPI51738.2020.00053..

[23] Mirkovic, Jelena, Gregory Prier, and Peter Reiher. "Attacking DDoS at the source." 10th IEEE International Conference on Network Protocols, 2002. Proceedings.. IEEE, 2002.

[24] J. Mirkovic, G. Prier, and P. Reiher, “Source-end DDoS defense,” in Second IEEE International Symposium on Network Computing and Applications, 2003. NCA 2003., pp. 171–178. doi: 10.1109/NCA.2003.1201153.

[25] S. I. Ao and International Association of Engineers., International MultiConference of Engineers and Computer Scientists : IMECS 2009 : 18-20 March, 2009, Regal Kowloon Hotel, Kowloon, Hong Kong. Newswood Ltd., 2009.

[26] X. Liang and T. Znati, “On the performance of intelligent techniques for intensive and stealthy DDos detection,” Computer Networks, vol. 164, Dec. 2019, doi: 10.1016/j.comnet.2019.106906.

[27] X. Wu et al., “Top 10 algorithms in data mining,” Knowledge and Information Systems, vol. 14, no. 1, Jan. 2008, doi: 10.1007/s10115-007-0114-2.

[28] D. Hu, P. Hong, and Y. Chen, “FADM: DDoS Flooding Attack Detection and Mitigation System in Software-Defined Networking,” Dec. 2017. doi: 10.1109/GLOCOM.2017.8254023.

[29] Z. Xie, W. Dong, J. Liu, H. Liu, and D. Li, “Tahoe,” in Proceedings of the Sixteenth European Conference on Computer Systems, Apr. 2021, pp. 426–440. doi: 10.1145/3447786.3456251.

[30] B. Charbuty and A. Abdulazeez, “Classification Based on Decision Tree Algorithm for Machine Learning,” Journal of Applied Science and Technology Trends, vol. 2, no. 01, pp. 20–28, Mar. 2021, doi: 10.38094/jastt20165.

[31] S. K. Murthy, “Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey,” Data Mining and Knowledge Discovery, vol. 2, no. 4, 1998, doi: 10.1023/A:1009744630224.

[32] H. Kousar, M. M. Mulla, P. Shettar, and D. G. Narayan, “Detection of DDoS Attacks in Software Defined Network using Decision Tree,” in 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT), Jun. 2021, pp. 783–788. doi: 10.1109/CSNT51715.2021.9509634.

[33] Gao, W. and Morris, T.H., 2014. On cyber attacks and signature based intrusion detection for modbus based industrial control systems. Journal of Digital Forensics, Security and Law, 9(1), p.3.